Merits of random forests emerge in evaluation of chemometric classifiers by external validation

Scott, Ian Morris; Lin, Wanchang; Liakata, Maria; Wood, J. E.; Vermeer, Cornelia Petronella; Allaway, D.; Ward, J. L.; Draper, John; Beale, M. H.; Corol, D. I.; Baker, J. M.; King, Ross Donald

Published in:
Analytica Chimica Acta

DOI:
10.1016/j.aca.2013.09.027

Publication date:
2013

Citation for published version (APA):
Scott, I. M., Lin, W., Liakata, M., Wood, J. E., Vermeer, C. P., Allaway, D., Ward, J. L., Draper, J., Beale, M. H., Corol, D. I., Baker, J. M., & King, R. D. (2013). Merits of random forests emerge in evaluation of chemometric classifiers by external validation. *Analytica Chimica Acta*, *801*, 22-33. [ACA232839]. https://doi.org/10.1016/j.aca.2013.09.027

Analytica Chimica Acta 801 (2013) 22–33

Contents lists available at ScienceDirect

Analytica Chimica Acta

journal homepage: www.elsevier.com/locate/aca

Merits of random forests emerge in evaluation of chemometric classifiers by external validation

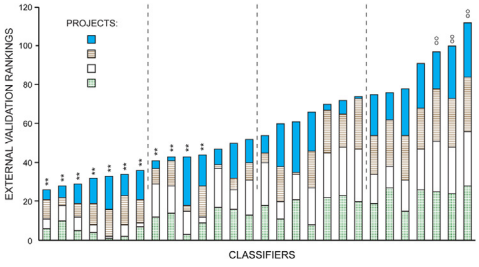
I.M. Scott[a,*], W. Lin[a,1], M. Liakata[b,2], J.E. Wood[a], C.P. Vermeer[a], D. Allaway[c,3], J.L. Ward[d], J. Draper[a], M.H. Beale[d], D.I. Corol[d], J.M. Baker[d], R.D. King[b,4]

[a] Institute of Biological, Environmental and Rural Sciences, Aberystwyth University, SY23 3FG, UK
[b] Department of Computer Science, Aberystwyth University, SY23 3DB, UK
[c] Mars Chocolate UK Ltd, Slough SL1 4JX, UK
[d] National Centre for Plant and Microbial Metabolomics, Rothamsted Research, Harpenden AL5 2JQ, UK

HIGHLIGHTS

- Only 6.6% of 286 reviewed papers clearly used 'external validation' on classifiers.
- We tested 28 classifiers on NMR or MS data of different origin to the training set.
- Data came from 4 metabolomics or food projects, whose class numbers differed.
- Random forests were best on high-dimensional data, but used in only 4.5% of papers.
- Feature selection with ReliefF improved other machine learning classifiers.

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 22 June 2013
Received in revised form 6 September 2013
Accepted 14 September 2013
Available online 23 September 2013

Keywords:
Classification
External validation
Machine learning
Prediction
Random forest
ReliefF

ABSTRACT

Real-world applications will inevitably entail divergence between samples on which chemometric classifiers are trained and the unknowns requiring classification. This has long been recognized, but there is a shortage of empirical studies on which classifiers perform best in 'external validation' (EV), where the unknown samples are subject to sources of variation relative to the population used to train the classifier. Survey of 286 classification studies in analytical chemistry found only 6.6% that stated elements of variance between training and test samples. Instead, most tested classifiers using hold-outs or resampling (usually cross-validation) from the same population used in training. The present study evaluated a wide range of classifiers on NMR and mass spectra of plant and food materials, from four projects with different data properties (e.g., different numbers and prevalence of classes) and classification objectives. Use of cross-validation was found to be optimistic relative to EV on samples of different provenance to the training set (e.g., different genotypes, different growth conditions, different seasons of crop harvest). For classifier evaluations across the diverse tasks, we used ranks-based non-parametric comparisons, and permutation-based significance tests. Although latent variable methods (e.g., PLSDA) were used in

Abbreviations: 9 × CV, nine-fold cross-validation; EV, external validation; FIE-MS, flow-injection electrospray-mass spectrometry; IID, independent and identically distributed; LDA, linear discriminant analysis; OSC, orthogonal signal correction; PCA, principal component analysis; PLSDA, partial least squares discriminant analysis; SD, standard deviation; SIMCA, soft independent modeling of class analogy.

* Corresponding author. Tel.: +44 1970 622341; fax: +44 1970 622350.
E-mail address: ias@aber.ac.uk (I.M. Scott).
[1] Present address: School of Medicine, University of Manchester, M13 9PT, UK.
[2] Present address: Department of Computer Science, University of Warwick, CV4 7AL, UK.
[3] Present address: WALTHAM Centre for Pet Nutrition, Melton Mowbray, LE14 4RT, UK.
[4] Present address: School of Computer Science, University of Manchester, M13 9PL, UK.

0003-2670/$ – see front matter
http://dx.doi.org/10.1016/j.aca.2013.09.027

64% of the surveyed papers, they were among the less successful classifiers in EV, and orthogonal signal correction was counterproductive. Instead, the best EV performances were obtained with machine learning schemes that coped with the high dimensionality (914–1898 features). Random forests confirmed their resilience to high dimensionality, as best overall performers on the full data, despite being used in only 4.5% of the surveyed papers. Most other machine learning classifiers were improved by a feature selection filter (ReliefF), but still did not out-perform random forests.

1. Introduction

1.1. Classifier validation in chemometrics

The technology-driven explosion of chemical data is fueling a widening range of expectations. Food analysts were among the first to exploit pattern recognition in chemical data of biological origin [1], followed by the now huge field of 'metabolomics' [2], with applications to human nutrition [3] and medicine [4] envisaged. Such ambitions demand robust data interpretation, but future real-world applications will inevitably include non-ideal data for chemometric analysis.

Given a set of samples (technical synonyms: 'objects', 'instances', 'observations') of known class, each described by a vector of chemical data (called 'features', 'variables', or 'attributes'), the aim of classification models ('classifiers') in chemometrics is to assign classes to new samples by pattern recognition in their chemical-feature vectors [1]. 'Supervised' classifiers are built on a 'training' population, with a priori knowledge of the class to which each sample belongs. The resultant model can be evaluated on 'validation' or 'test' samples, whose class membership is unknown to the classifier. The concept appears statistically straightforward if the training and validation samples come from independent and identically distributed (IID) populations [5].

In this paper, we confront the more realistic and challenging scenario that chemometric classifiers will be queried using novel data populations, in which the IID assumption may be violated. Relative to training data, test samples may be generated at different times, under different conditions, be different in nature, contain different class proportions, and so on. Analogous problems are recognized in other areas of pattern recognition, such as text mining and computer vision; indeed, algorithms for non-IID data are an emergent theme in data mining [6].

Capture of real-world data scenarios in chemometrics papers has been rare, due to limitations of sample availability [1,7]. This has caused disputes over the merits of compromises for validation, with different fields offering divergent perspectives [5,8]. A long-standing practice has been to split data randomly into one part for model building, and the remainder held out for model testing [8]. When surveying methodologies used in the present study, we collated 286 chemometrics papers, of which 41% included single-split hold-outs (Table 1). Among these, 8.7% incorporated a selection constraint or algorithm, such as Kennard–Stone [9], to maximize the modeling potential of single subsets, though random selection was more common (Table 1).

Modern statisticians counsel against single-split hold-outs for data sets whose limited size means splitting may be detrimental to model generalizability [8]. Instead, resampling is now favored, and its most popular form in chemometrics (58% of papers, Table 1) is cross-validation (CV). In theory, CV maximizes the utility of the available data, by using the entire sample population for both model-building and validation [5,8]. This usually involves successive data partitions, such that in each round a fraction is withheld to test a model trained on the remaining data. 'Full CV' averages performance over all data partitions when every sample has been withheld and predicted once. Other forms of resampling, such as permutation and bootstrapping, are widely used in statistics [5,8], but featured in only 5.9% of our surveyed chemometrics papers (Table 1).

Table 1
Validation strategies in 286 classification papers (2002–2012) from five journals (*Anal. Chem.*, *Anal. Chim. Acta*, *Chemom. Intell. Lab. Syst.*, *Food Chem.*, *Metabolomics*).[a]

Validation	Percent of studies [b]
Cross-validation	58.4
Random hold-out	23.4
Autopredictive	9.4
Designed hold-out [c]	8.7
Undefined hold-out [d]	8.4
Explicitly independent test set [e]	6.6
Permutation	3.5
Bootstrap	2.4

[a] Data from census of papers using methodologies tested in the present study (see Table 7).
[b] Categories not mutually exclusive, e.g., 16.4% used both cross-validation and hold-out.
[c] Training/test splits using algorithms (Kennard–Stone, D-optimal, or Kohonen self-organizing maps), or sampling constraints (e.g., variation range, class balance).
[d] Presumably random, but no method statement on training/test split.
[e] Stated difference in provenance of training and test populations.

The resampling paradigm has critics, who argue that all forms of data-splitting and resampling, including CV, are inherently sub-optimal simulations of 'external validation' (EV) with an independently obtained test set [5]. Only the latter, it is argued, can realistically reflect sampling errors in data acquisition [5]. Testing classifiers with such 'external' sample populations was more challenging than CV in recent studies where we built regression models [7], or used a limited range of classifiers [10].

We therefore reviewed the 286 papers of Table 1 for stated elements of variance between training and test samples. Examples included: cork [11], olive oil [12], clinical cream [13], or illegal cannabis [14], from different sources; wines produced in different years [15]; aspen leaves grown in different conditions [16]; different viral subtypes [17]; or urine collected on different days [18]. In total, however, only 6.6% of papers qualified for this category (Table 1).

This estimate may be conservative, if authors understate the independence of their samples, but clearly the challenges of EV and non-IID data have not been a priority in chemometrics. Subjects for EV research can only expand as laboratories accrue data on nominally comparable samples whose production will inevitably vary over time. The present study took advantage of spectrometric data generated over several projects. Data sets were deliberately identified as being non-ideal in some respect for classification purposes. They were used as a testing space to identify the better classifiers, of the many now available, for the challenges of EV.

1.2. Methodological background

Both classical 'latent variable' [19], and 'machine learning' methods [20] were evaluated using class-labeled data examples to construct empirical models for classification of further data. Machine learning used the Weka package, whose terminology and categorization for its algorithms [20] are followed in the summary provided in Table 2.

Table 2
Overview of classification methods evaluated.

Category	Scheme[a]	Classification principle
Latent variables	SIMCA	'Class modeling' by distance to independent PCA models of each class
	PCA-LDA	Transformation by linear combination of PCs that best separates classes
	PLSDA	Models latent variables for maximal correlation to classes
	OSC-PLSDA	Prior to PLSDA, variation orthogonal to class is subtracted
Bayesian	*NaiveBayes*	Standard probabilistic Bayesian classifier[b]
Functions	*MultilayerPerceptron*	Backpropagation neural networks
	RBFNetwork	Radial basis function networks
	SimpleLogistic	Linear logistic regression with built-in feature selection
	SMO	Support vector machines[b] with sequential minimal optimization
Nearest neighbors	*IBk*	*k*-Nearest neighbors[b]
	NNge	Nearest-neighbors using hyperrectangles of if-then rules ('nonnested generalized exemplars')
Simple	*HyperPipes*	Hypervolumes in sample space
Decision trees	*J48*	C4.5 decision trees[b]
	RandomForest	Ensemble of decision trees built on random features from bootstrapped data
	SimpleCart	Classification/regression trees[b]
Feature evaluation	*ReliefF*	Heuristic sampling to weight features using nearest samples of same and different classes
Search methods	*Ranker*	Ranks features by individual evaluations

[a] Weka schemes are in italics, in categories corresponding to their Weka subclass [20].
[b] Voted among 'top ten' most influential data mining algorithms at 2006 IEEE conference [25].

1.2.1. Latent variable classifiers

The rationale of latent variables is that certain factors may cause correlated behavior among features of a sampled material during its genesis. A new description of multicollinear features as composite variables reduces dimensionality, and creates interpretive opportunities.

The foundational latent variable method, principal component analysis (PCA), classically finds the eigenvectors of the data covariance matrix, and ranks these by their eigenvalues. Projection of the original data onto the highest-ranked eigenvectors reveals the principal components (PCs) encapsulating most data variance [19].

PCA is 'unsupervised', involving no reference to data classes, but our survey included supervised applications of PCs. One was 'soft independent modeling of class analogy' (SIMCA), which, as a 'class modeling' technique, can return a non-result, i.e., samples in no defined class [21]. In SIMCA, training data of each class are used independently to build PC models, into which test samples are projected. Test samples are assigned to a class if they fall within the critical distance to the scores range of the class model.

A classical pattern recognition method is linear discriminant analysis (LDA), which obtains latent variables in the form of linear combinations of the original data features that maximize between-class, and minimize within-class, variance [22]. Since LDA is subject to the constraint that the number of features should not exceed the number of samples, our high-dimensional spectral data were reduced to PCs prior to LDA.

For high-dimensional data, another solution to the above constraint of LDA is partial least squares discriminant analysis (PLSDA). This seeks components that describe the variance in the sample features matrix having maximal correlation with known class values, giving less weight to class-irrelevant or noise variance. Although PLSDA emerged independently of LDA, the two have similar underlying statistical constructs [23]. Broadly similar performances might therefore be expected for PLSDA and PCA-LDA, but as each is widely employed in chemometrics, we tested both.

In a correction routine developed for spectroscopy [24], PLS components weighted to minimize covariance between features and class can be removed from the data matrix prior to PLSDA. This orthogonal signal correction (OSC) is supposed to improve classification of latent variable models. We examined how OSC prior to PLSDA affected classification of independent test data.

1.2.2. Machine learning classifiers

One should first try simple algorithms before resorting to more complex solutions [20]. We therefore included as a baseline Weka's little-used *HyperPipes*. This records the range of values for each data feature and class in the training examples, and test samples are assigned to classes that contain the largest amount of matching ranges.

We also tested simple classifiers, which, though not reputedly the most powerful in chemometrics, were voted among the data mining community's 'top ten' most influential algorithms at the 2006 IEEE International Conference on Data Mining [25]. These included Naive Bayes and *k*-nearest neighbor algorithms. Naive Bayes is among the oldest classification algorithms, important in fields outside chemometrics [25,26]. It builds a probabilistic model with the 'naive' assumption that features within a class are mutually independent [20,25]. Weka's default *NaiveBayes* assumes data are normally distributed. Non-parametric approaches include *k*-nearest neighbor approaches [20]. The default *IBk* algorithm in Weka assigns an unknown sample to the class of the neighbor that is nearest by Euclidean distance [20]. It is thus strictly 'instance-based', using training examples without a generic model. The Weka toolkit also has *NNge*, a hybrid of instance-based classification that models 'generalized exemplar' hyperrectangles, whose dimensions cover a set of training examples [20].

Two more algorithms among the most influential in data mining are the decision tree classifiers C4.5 and CART [25], implemented in Weka as *J48* and *SimpleCart*, respectively. Both construct branched 'trees', with nodes representing successive splits of the data by values of class-discriminating features. Splits in CART are binary, whereas C4.5 allows more outcomes. Criteria by which the two algorithms evaluate splits, and 'prune' fully-grown trees, also differ [25]. By the late-1990s, decision trees were comparing poorly with emergent competitors like support vector machines. Decision tree methodology was subsequently improved by the development of random forests [27]. These construct models comprising a 'forest' of decision trees, each formed on a random subset of features, from a random subset of samples, selected (with bootstrapping) from the training population. Test samples are evaluated by every tree in the forest, and classification decided by their consensus. The aggregate predictions of these decision tree forests compare well to many other methods [27].

Support vector machines were our fifth representative from the 'top ten' algorithms [25]. The 'support vectors' of the training data set are the opposite-class examples with least mutual separation, and are used to find a separating 'hyperplane' equidistant from each class. Linear separation in higher dimensions is achieved by the use of 'kernel' functions [1,25]. Weka's implementation, *SMO*, uses the sequential minimal optimization algorithm [20].

SMO is among classifiers in Weka's *functions* subclass, whose models could be written as mathematical equations [20]. Historically pre-eminent among these are artificial neural networks [28], of which two architectures were tested: *MultiLayerPerceptron* implements back-propagation neural networks, and *RBFNetwork* normalized Gaussian radial basis function networks [20].

Also in this Weka subclass is *SimpleLogistic*, a 'boosting' algorithm. It fits linear regression models step-wise, each using the best remaining data feature. CV is used to determine the optimal set of simple linear regression models, which are finally combined in a logistic regression classifier [20,29].

1.2.3. Feature selection

Data 'dimensionality' in analytical chemistry reflects the number of elements in chemical vectors, which can be thousands. High-dimensional data spaces have insidious properties that can confound their representational power [30]. Spurious class-correlations of individual features are more likely in high-dimensional training data, and if incorporated by classifiers, these may not generalize to new examples (so-called 'overfitting').

Our tested classifiers varied in reputed suceptibility to dimensionality. Algorithms that use a distance measure are generally vulnerable to the dilution of class-discriminative dimensions by numerous irrelevant ones; this applies to *k*-nearest neighbor [30] and neural network approaches [28]. Decision trees are also vulnerable, if their branches proliferate excessively as the number of data features increases [30]. The susceptibility of Naive Bayes to dimensionality is data-dependent. It is robust to irrelevant features but, as it multiplies probabilities for individual features, it can be biased by interdependent features [20].

Support vector machines cope well with high dimensionality, due to controls on model complexity, and maximization of generalizability by focus on support vectors. However, they can benefit from reduction in dimensionality [30]. Random forests avoid overfitting by aggregating many relatively low-dimensional classifiers with low inter-correlation [27].

Reduction of dimensionality with selection of class-correlated features may therefore be beneficial in machine learning [30]. Classifiers susceptible to over-fitting may be improved by a prior feature selection algorithm as a 'filter' [30]. We therefore tested the Weka 'attribute evaluator' *ReliefF*, which is an instance-based algorithm that repeatedly picks a random sample from the training population, and weights every feature by discrimination of nearest neighbors of the same and different classes [31]. In one of our tested classifiers, *SimpleLogistic*, feature selection is 'embedded' via its step-wise selection of simple regression models [29].

1.2.4. Classifier comparison statistics

Our objective was to evaluate classifiers simply for correct class prediction. It should be mentioned that other model properties, such as complexity or interpretability, were outside our scope, but might figure in the choice of classifiers for particular purposes. Several diagnostic statistics exist, moreover, for simply quantifying predictive ability [32]. We used the intuitive criterion of 'accuracy' (here, percent of predictions that are correct), which usually features as at least one of the metrics in classification studies. The merits for PLSDA model optimization of such a simple metric have recently received empirical support [32].

The different numbers and prevalence of classes in the four projects meant, however, that expected rates of correct predictions could differ due to random chance [33]. We therefore sought to supplement 'accuracy'-based evaluations with a metric insensitive to class distributions. Several standard measures ('recall', 'precision', 'F-measure') derived from tallies of true/false (T/F) and positive/negative (P/N) predictions [20,34], would be subject to class numbers and prevalence. A more robust solution would be to evaluate the trade-off between TP and FP rates, via 'receiver operating characteristic' curves [20,34]. Although these are widely used for binary (two-class) classifiers, however, their attraction was limited by the lack of a consensus protocol for multiple class evaluations [35].

The extra metric we chose for better comparisons across differing numbers and balance of classes was Cohen's kappa, which is designed to deduct the portion of success rates attributable to chance [20,33,34]. Kappa has been less used in data mining than in other fields [33], though it is one of Weka's standard measures [20]. It is defined as

$$\kappa = \frac{P_o - P_c}{1 - P_c}$$

where P_o = predictor's observed success rate, and P_c = predictor's chance success rate (empirically estimated from confusion matrices) [20,33,34]. Since kappa has its own limitations [34], we adopted it as a supplement, rather than a replacement of the accuracy measure.

Our evaluation space comprised NMR or mass spectrometric data sets, from four projects with different metabolomic or food production objectives. The four projects differed in class numbers (from two to five), and in absolute and relative numbers of samples within each class. Within each project, there were distinct subsets of data. Most importantly, between these subsets there were sources of variation in the production or genetics of the relevant materials. Training and test data in EV were represented by distinct subsets, and hence differentiated, potentially non-IID populations. Inter-project statistical comparisons of classifier performances needed to be robust to the potentially widely different challenges presented by each project.

Demšar [36] argued, in 2006, that statistical tests for comparisons of multiple classifiers over multiple data sets were largely unexplored, and established procedures were lacking. For parametric tests, comparisons of algorithms on multiple data sets pose several problems: the data sets may be incommensurate; normality and homogeneity of variance/sphericity may be violated; and outliers may skew the statistics [36]. Theoretically and empirically, Demšar [36] recommended the non-parametric Friedman test. This is a ranks-based 'repeated-measures' test, involving comparisons of multiple observations (here, classifier performances) on the same series of subjects (here, data sets). The Friedman test can be seen as a non-parametric counterpart of the repeated-measures ANOVA, or a repeated-measures counterpart of the Kruskal–Wallis test. Differences between average ranks of our classifiers were evaluated for significance using the critical difference of the post hoc Nemenyi test [36]. The merits of ranking for method comparison in analytical chemistry have also been advanced by Héberger [7,37].

Performance rankings were therefore adopted here as a robust measure for unified comparisons of classifiers over diverse projects. This exercise would have limited value, however, unless at least some classifiers were demonstrably successful. Again, a measure robust to the incommensurability of the different projects was desirable. For such purposes, permutation tests have been advocated [38]. Their null hypothesis is that performance is no better than random chance. *P*-values can be assigned as frequencies with which accuracies on randomized data match those on real data [38].

Using this statistical toolkit, we evaluated the absolute and relative performances of a suite of classifiers in validations more challenging than most of the current literature.

2. Materials and methods

2.1. Experimental designs and data acquisition

A summary of the four projects and their data sets is in Table 3.

2.1.1. 'Invertases' project

The two classes were 'wild-type' versus 'invertase-mutant' phenotypes of the *Arabidopsis thaliana* Col-0 line, to be identified from NMR spectral fingerprints of shoots. The putative source of variation between sample sets was that the mutant phenotypes were affected in different invertase genes. *A. thaliana* has genes for several invertases (sucrose-hydrolyzing enzymes), which differ in cellular location and apparent roles in plant development [39–41]. Mutants of the following genes were fingerprinted: the mitochondrial, alkaline/neutral invertases *At-A/N-InvA* (At1g56560) and *At-A/N-InvH* (At3g05820); a cell wall, acid invertase *AtcwINV5* (At3g13784); the vacuolar, acid invertases *Atβfruct3* (At1g62660) and *Atβfruct4* (At1g12240), plus a double mutant of these last two [39–41]. The morphology of the mutants did not appear to be abnormal.

Nine replicate *A. thaliana* plants of Col-0 and each of the six mutants were grown in a single experiment (in 250–270 μmol m^{-2} s^{-1} light) and harvested as in Scott et al. [10]. Shoot extracts were prepared and NMR spectra acquired on a Bruker Biospin Avance spectrometer (Coventry, UK) at 600 MHz [10].

2.1.2. 'Biomass' project

The three classes were wild-type (Col-0), salicylate-deficient or salicylate-overproducer genotypes of *A. thaliana*, which were to be identified from flow-injection electrospray-mass spectrometry (FIE-MS) fingerprints of shoots grown at chilling (5 °C) temperature. The genetic 'background' of all was the Col-0 line. Those deficient in salicylate were either mutant in *sid2* (At1g74710) or *eds5* (At4g39030), or transgenic for *NahG*. The overproducer was mutant in *cpr1* (At4g12560). Under chilling conditions, shoot biomass in *A. thaliana* has been found to be inversely proportional to salicylic acid content [42].

The putative source of variation between sample sets was that they came from plants grown at 5 °C in four batches for different periods of time (42–100 days), and in different light levels (25–100 μmol m^{-2} s^{-1}). Shoot biomass averages relative to Col-0 varied in the four batches, from 1.33 to 3.02 × Col-0 in the salicylate-deficient plants, to 0.06–0.25 × Col-0 in the salicylate-overproducer plants.

Shoots were extracted for analysis by FIE-MS on a Bruker Esquire 3000 spectrometer (Coventry, UK), by the methods of Ward et al. [43]. Replicates varied numerically, as the tiny *cpr1* mutants required bulking to achieve sufficient analytical material [42]. Moreover, as there were three salicylate-deficient genotypes, this class was numerically over-represented in the data sets.

2.1.3. 'Starch/lipid' project

The four classes were the wild-type and three mutants of the Col-0 line of *A. thaliana*, to be identified from FIE-MS spectral fingerprints of shoots. The mutants were affected in metabolism of starch (a phosphoglucomutase encoded by the *pgm* gene, At5g51820) or lipid (a plastid-localized glycerol-3-phosphate acyltransferase encoded by *ats1*, At1g32200; and an endoplasmic reticulum-localized 18:1-phosphatidylcholine desaturase encoded by *fad2*, At3g12120) [10]. The morphology of mutant plants did not appear to be affected.

The putative source of variation between sample sets was that they came from plants grown in three batches in either 100–150 or 250–270 μmol m^{-2} s^{-1} lighting, as described by Scott et al. [10].

Table 3
Overview of materials and classes of the four projects.

Project	Material	Spectral data	Number of spectral variables	Classes	Ratios of samples per class	Sources of variation	Mean number of replicates ± SD
Invertases	*Arabidopsis* shoots	NMR (7 sets)	914	Wild-type, invertase mutant (2 classes)	1:1 (all sets)	Mutants of different genes	9 ± 0[a]
Biomass	*Arabidopsis* shoots	FIE-MS (4 sets)	1898	Wild-type, high-biomass mutant, low-biomass mutant (3 classes)	28:5:4; 16:9:9; 30:10:10; 16:5:5	Different growth conditions	12.2 ± 8.9[a]
Starch/lipid	*Arabidopsis* shoots	FIE-MS (3 sets)	1852	Wild-type, *pgm* mutant, *ats1* mutant, *fad2* mutant (4 classes)	1:1:1:1 (all sets)	Different growth conditions	8.7 ± 0.5[a]
Cocoa	Fermented cocoa beans	FIE-MS (3 sets)	1872	Crop varieties (5 classes)	1:1:1:1:1 (all sets)	Crops from different seasons	35.0 ± 14.1[b]

[a] Replicates were distinct biological materials.
[b] Replicates were analytical samples from a single fermentation batch.

Eight or nine replicate plants were grown in each batch. Shoots were harvested and analyzed by FIE-MS on a Waters Micromass LCT spectrometer (Elstree, UK) as in [10]. We had already noted that random forests were able to recognize the spectral fingerprints of these mutants from one growth batch to another [10].

2.1.4. 'Cocoa' project

The five classes were varieties of *Theobroma cacao*, to be identified after bean fermentation, which is a key stage in cocoa production [7]. Two were traditional cultivars: Amelonado, commonly grown in South America and West Africa; and EET53, a 'Nacional' type from Ecuador. Another was Scavina 6, a disease-resistant clone collected from the upper Amazon. These three probably belonged to genetically distinct groups [44]. The others were: Catongo, an Amelonado type with white beans; and CCN51, a disease-resistant variety of complex pedigree widely planted in Ecuador [7].

The putative source of variation between sample sets was that they came from beans harvested in three different seasons (2001, 2002, 2003), on a plantation at Fazenda Almirante (a division of Mars, M&M Inc. in Itajuipe, Bahia, Brazil), where they were fermented as previously described [7].

Chemical extraction and FIE-MS analysis on a Waters Micromass LCT was as described by Wood et al. [7]. Each season's production of a given variety formed a single sample, which was subdivided for analytical replication.

2.2. Classification of data

2.2.1. Spectral data characteristics and pre-processing

NMR spectra (Invertases project) were acquired in 128 scans of width 7310 Hz, and Fourier-transformed with an exponential window (0.5 Hz line broadening). Spectra were binned to 0.01 ppm, and intensities scaled relative to the chemical-shift reference peak ($^2H_4$-trimethylsilylpropionate, δ0.05 to −0.05). The output data had 914 features in the range δ9.995–0.505. Three analytical replicates were averaged.

FIE-MS fingerprints (Biomass, Starch/lipid, and Cocoa projects) were acquired in ranges between *m/z* 51 and 1000 (depending on the project), binned to unit *m/z* values, and normalized to total ion current of the sample infusion [45]. Positive- and negative-ion spectra were concatenated, yielding 1852–1898 features (Table 3).

Any additional data processing steps prior to the various classification routines are in the Supplementary Data (Table S1).

Supplementary material related to this article can be found, in the online version, at http://dx.doi.org/10.1016/j.aca.2013.09.027.

2.2.2. Latent variable approaches

Latent variable analyses were applied to mean-centered data, without scaling. The R packages *stats* and *MASS* [46] were used for PCA-LDA, while SIMCA-P version 11.0 (Umetrics AB, Umeå, Sweden) was used for SIMCA, PLSDA and OSC-PLSDA. The latter software follows a 'PLS2′ procedure, in which all classes are modeled simultaneously. When OSC was used, class-orthogonal components with eigenvalues > 2 were removed from models [24]. In SIMCA-P, the number of components to retain in models is determined by a CV process [47]; the default of seven data subsets was used in this CV for all methods (including PCA-LDA, for which PCA in SIMCA-P was performed to determine the PCs to retain).

2.2.3. Machine learning classifiers

Classifiers employed via the Explorer interface of the Weka machine learning workbench, version 3.6.1 [48], included: *NaiveBayes*, *MultilayerPerceptron*, *RBFNetwork*, *SimpleLogistic*, *SMO*, *IBk*, *NNge*, *HyperPipes*, *J48*, *RandomForest*, and *SimpleCart*. *SMO*, which is a support vector machine implementation, was performed with both a linear kernel (*PolyKernel* exponent = 1.0), and a second-order polynomial kernel (*PolyKernel* exponent = 2.0). Weka performs multiclass *SMO* classifications using pairwise analyses.

To reduce the danger of overfitting, model parameters were not tuned during training [7]; instead a single fixed set of parameters was used, as detailed in the Supplementary Data (Table S1). We generally preserved defaults of the Weka package, though for *RandomForest* we grew 1000 trees, using the default number of random features ($\sqrt{m}$, where m = number of data features) of the original algorithm [27].

Reduction of data dimensionality was performed in the Weka Explorer using the *ReliefF* feature evaluation method, with the *Ranker* search, to select 100 features prior to application of each Weka classifier. The feature evaluator/search method is referred to as a 'filter' [31], and the classifier without it as the 'base' classifier. Samples destined for classifier testing were completely excluded from the feature selection process [49].

2.3. Analysis of classifier performances

2.3.1. Validation designs

Full, nine-fold cross-validations (9 × CV) were performed on all pairs of the sets of spectral data (Table 3) in each project. The pooled data were partitioned in each round such that one-ninth of samples were withheld and used to test the predictive accuracy of the model. Selection of the held-out samples was random within classes, while seeking to reflect the population class distribution. This was iterated until every sample in the population had been withheld and predicted once. The held-out subsets were excluded from all stages of model building, including optimization of model components [50]. 9 × CV was performed separately for each of the classifiers, so these were not tested with identical data subsets.

The EV procedure differed from 9 × CV in that sources of variation (e.g., different plant growth conditions, harvest years, or genotypes) were explicitly partitioned between the 'training' data used to build models, and the data on which models were tested. Only a single data set was used for training. For example (Table 4), projects with three sets of data (e.g., the three seasons of the Cocoa project) would entail six rounds of EV, as the classifier was trained on each season's data and tested on data from each of the other two. The Biomass project with four data sets (plants grown at different times in varying conditions), entailed 12 rounds of EV. For the Invertases project, training involved one mutant and the wild-type Col-0, with testing on each of the other five mutants (30 EV rounds).

2.3.2. Classifier performance measures

Accuracy and Cohen's kappa statistic were used as metrics of performance. In a given classification task, accuracy was defined as the percentage of times class was correctly predicted, irrespective of the numbers of samples or classes. In Table 4 illustration, accuracy for classification task (1) would be the percentage of Set 2 samples whose class was correctly predicted, accuracy for task (2) would be the percentage of correctly classified Set 3 samples, and so on.

Kappa was calculated as described by Ben-David [33] except for special cases where this was inapplicable [34]. One was the Invertases data, where only one of two possible classes populated the test sets, and here P_c was taken as 0.5. The other was for SIMCA class-modeling when predictions were inconclusive; here, kappa was arbitrarily set to 0.

2.3.3. Significance of classification performances

Permutation tests [38] were used to estimate the statistical significance of classification accuracies. Training and test data sets for

Table 4
Example EV design for project with 3 sets of data: the classifier is trained on each set, and tested on each of the other two sets.

(1) Train/test	(2) Train/test	(3) Train/test	(4) Train/test	(5) Train/test	(6) Train/test
Set 1/Set 2	Set1/Set 3	Set 2/Set 1	Set2/Set 3	Set 3/Set 1	Set3/Set 2

a given classification task were replicated 1000 times with all class-labels randomly permuted (with the R *base* package). *P*-values were defined as the proportion of times accuracies on real data were equaled or surpassed in the 1000 tests on permuted data. Using Table 4 as illustration, 1000 permutations of the 'Set 1/Set 2' data would be generated to obtain *P*-values for train/test task (1), then 1000 permutations of the 'Set 1/Set 3' data would be generated to obtain *P*-values for train/test task (2), and so on for all six tasks. All *P*-values were based on 1000 permuted data sets, irrespective of sample or class numbers.

For Weka classifications, permuted data were written to ARFF files using the *RWeka* package [51], and batch-processed via the Weka Experimenter interface [48]. PCA-LDA used the R packages *stats* and *MASS* [46], and PLSDA the R package *plsgenomics* [52], on permuted data. For both, the number of components modeled was (arbitrarily) determined by SIMCA-P on the unpermuted data (see Section 2.2.2). OSC [24] prior to PLSDA on permuted data was performed in R, the components removed equaling the number determined by SIMCA-P on the unpermuted data (see Section 2.2.2).

2.3.4. Comparative performances of classifiers

For a given project, several metrics were used to summarize the overall performance of each classifier. These were the average accuracies and kappa values (see Section 2.3.2), and average percentages of significant classifications (see Section 2.3.3) over all the training/test set combinations (see Section 2.3.1) in the project. For illustration, the EV design in Table 4 would generate six accuracy (or kappa) values, one for each of tasks (1)–(6). The arithmetic mean of these six values would represent the classifier's average accuracy (or kappa score) for that project.

Performances of all classifiers in the project were ranked by these metrics. Statistical comparisons of these classifier rankings over all four projects used the non-parametric, ranks-based, Friedman test (in the R *stats* package), with post hoc Nemenyi tests [36]; we refer to these as 'Friedman–Nemenyi' tests.

In addition, we quantified each classifier's performance relative to the highest known potential performance, exemplified by whichever classifier was best overall in the same project. This measure was the ratio between the average accuracy of the given classifier and that of the best-performing one.

Pairwise comparisons of classifiers over multiple tasks used Wilcoxon signed-ranks tests, as advocated by Demšar [36], with Monte Carlo *P*-values, in PAST version 1.91 [53].

3. Results

3.1. Classifiers evaluated

Twelve machine learning classifiers were evaluated, these being the Weka algorithms *NaiveBayes*, *MultilayerPerceptron*, *RBFNetwork*, *SimpleLogistic*, *SMO* with a linear kernel, *SMO* with a second-order polynomial kernel, *IBk*, *NNge*, *HyperPipes*, *J48*, *RandomForest*, and *SimpleCart*. These were also tested using Weka's *ReliefF* feature selection filter. Four latent variable methods were tested: SIMCA, PLSDA, OSC-PLSDA, and PCA-LDA. Thus, 28 classifiers were evaluated.

3.2. Cross-validation versus external validation

We conducted 9 × CV versions of the classification tasks outlined in Section 2.1, alongside more challenging EVs. In 9 × CVs, sources of variation (Table 3) were pooled in a pairwise fashion, i.e., two invertase-mutants, two growth experiments, or two seasons. In EVs, sources of variation were explicitly separated as training and test sets. Within each project, EV involved every data set in turn being used for training, while each of the others served for testing (Table 4). Table 5 summarizes 27 9 × CV and 54 EV performances of 28 classifiers.

Comparison of 9 × CVs and EVs confirmed the latter were a greater challenge (Table 5). For three projects, at least one classifier achieved perfect classifications in 9 × CV, while average 9 × CV accuracies of all 28 classifiers were 75–95%. (We should note that 9 × CV lacked rigor for the Cocoa project, which involved only 'analytical' replication, but the results are included to show the data characteristics.)

By contrast, in EVs mean accuracies of the 28 classifiers were only 48–76%. That said, the best-performing EV classifiers were 86–97% accurate, confirming classifiable structure in the data (Table 5). As the projects differed in class numbers and balance, kappa scores were also estimated (Table 5). Kappa scores for the five-class Cocoa tasks were higher than for the two-class Invertases tasks, even where percent accuracies would not indicate this relative performance. Accuracy and kappa scores both confirmed the class-imbalanced Biomass data sets were relatively well classified.

3.3. Effects of feature pre-selection

All data were high-dimensional, with 914–1898 features. For reduction to fewer, class-predictive features, we tested a 'filter' on

Table 5
Comparison of cross-validation and external validation performances of 28 classifiers.[a]

Project	Mean % accuracies ± SD (mean kappa in brackets)			
	Cross-validations[b] (9 × CV)		External validations[c] (EV)	
	Best classifier(s) per test[d]	All classifiers[e]	Best classifier(s) per test[d]	All classifiers[e]
Invertases	88.5 ± 3.4 (0.76)	75.5 ± 9.2 (0.44)	86.2 ± 5.7 (0.71)	58.5 ± 12.2 (0.21)
Biomass	100 ± 0 (1.0)	94.5 ± 5.7 (0.88)	96.9 ± 6.7 (0.97)	75.5 ± 15.2 (0.58)
Starch/lipid	100 ± 0 (1.0)	91.0 ± 9.9 (0.88)	91.0 ± 7.2 (0.89)	62.5 ± 19.4 (0.52)
Cocoa	100 ± 0 (1.0)	95.0 ± 8.0 (0.94)	87.0 ± 10.8 (0.84)	48.4 ± 17.0 (0.36)

[a] Classifier list in Section 3.1.
[b] Tests per project: invertases (n = 15); biomass (n = 6); starch/lipid (n = 3); cocoa (n = 3).
[c] Tests per project: invertases (n = 30); biomass (n = 12); starch/lipid (n = 6); cocoa (n = 6).
[d] Means of top performance (by any classifier) in each test.
[e] Means of all 28 classifiers in each test.

Table 6
Effects of feature selection on EV performances of 12 classifiers.[a]

Project	Base classifier	*ReliefF* (top 100)
Mean accuracy ± SD (%)[b]		
Invertases	62.1 ± 9.1	60.9 ± 9.4
Biomass	77.1 ± 8.3	80.9 ± 5.4
Starch/lipid	56.3 ± 14.9	75.1 ± 9.3**
Cocoa	45.2 ± 14.0	57.5 ± 12.5**
Mean kappa ± SD[b]		
Invertases	0.26 ± 0.21	0.22 ± 0.19
Biomass	0.55 ± 0.18	0.67 ± 0.09*
Starch/lipid	0.43 ± 0.20	0.67 ± 0.12**
Cocoa	0.30 ± 0.18	0.46 ± 0.16**

[a] The Weka classifiers listed in Section 3.1.
[b] Mean performances of 12 classifiers in 6–30 tests per project.
Asterisks indicate improvement due to feature selection:
* $P < 0.05$;
** $P < 0.01$ (Wilcoxon tests).

the 12 machine learning classifiers. The application of *ReliefF* [31] in selection of 100 features frequently improved EV performances, according to Wilcoxon tests on accuracy and kappa scores (Table 6).

3.4. Comparative evaluations of classifiers

3.4.1. Performance rankings

Rankings are a robust measure, and underly many non-parametric tests [36]. We used cumulative mean rankings (over all four projects) to rate performances for 9 × CV (Fig. 1A) and EV (Fig. 1B), with Friedman–Nemenyi tests [36] to determine which classifiers differed significantly from the top- and bottom-ranked ones.

The overall performances of classifiers with a *ReliefF*-filter confirmed the generally beneficial effects of feature pre-selection already noted in Table 6. The only base classifier in both 9 × CV and EV top groups (i.e., not differing in Friedman–Nemenyi tests from the highest-ranked classifier) was *RandomForest*. Also in both top groups were *ReliefF*-filtered *RandomForest*, *SMO*, and *MultilayerPerceptron*. Conversely, the decision tree classifiers *SimpleCart* and *J48* were in the bottom quartile in both validations.

Each classifier was implemented with a single, fixed set of parameters for all classification tasks (Supplementary data, Table S1). Regarding machine-learning classifiers with tunable parameters, we should mention that the chosen fixed parameters gave 100% classification rates on all EV training sets for *SMO*, and for 99.7% of EV training sets for *MultilayerPerceptron*.

No latent variable classifiers were top quartile in either validation (Fig. 1). PLSDA was second quartile in 9 × CV, but all latent variable classifiers were fourth quartile in EV. OSC appeared detrimental to PLSDA. Rankings of OSC-PLSDA among the 28 classifiers were poorer than PLSDA in 9 × CV and EV (Friedman–Nemenyi tests; $P < 0.01$).

An issue for latent variable classifiers is the number of components to use in models [22]. The results shown for all four latent variable methods were obtained from models whose retained components were determined using the default CV in SIMCA-P software. Alternatives were, however, explored. No significant differences to the reported EV performances of PCA-LDA were found when retained components were fixed as ten, or determined by 9 × CV on training data. The maximum number of PCs available for LDA [22] was on average 23.1 (SD, 10.3), and by consistently using all these, PCA-LDA would have been promoted to third quartile in the EV rankings.

SIMCA class-modeling ranked as poorest of all methods (Fig. 1). We could not improve SIMCA by alternatively scaling the mean-centered data by unit variance ($P > 0.05$; Wilcoxon tests). It could be argued, however, that our performance metrics were not entirely commensurate for SIMCA. As a one-class classifier [54], for example, its null classifications obviated chance 'hits'. Though this was not the intention of its originators, it is possible to force SIMCA to assign samples to whichever class is closest, irrespective of the critical distance [54]. By this alternative procedure, the mean EV accuracy over all data sets increased to only 38.3%, which left SIMCA ranked in bottom place. For a couple of projects, however, forced-classification promoted SIMCA in the EV accuracy rankings, to 21st place for the Starch/lipid data, and 27th place for the Cocoa data.

3.4.2. Percentages of significant classifications

Classifier rankings would have little merit for non-significant classifications. For statistical evaluation, we ran dummy classifications with 1000 random class-label permutations [38]. This was done for all 54 classification tasks, each with 28 classifiers. Percentages of EV classifications that were significant are shown in Fig. 2A. The highest ranked classifiers were *ReliefF*-filtered *NaiveBayes* and *HyperPipes*, but *RandomForest* did not differ significantly from these (Friedman–Nemenyi tests; $P > 0.05$).

3.4.3. Minimum performances relative to best classifier

A classifier may be judged poor if it fails despite better performances by alternative classifiers. We therefore expressed each classifier's average performance in EVs for each project, as a proportion of the best classifier's performance on the same data. Fig. 2B presents each classifier's *worst* performance, in any of the four projects, by this metric. *RandomForest* had the best minimum performance (at 0.86 of the top classifier for the Cocoa data, which was *SimpleLogistic*). The next best nine were all *ReliefF*-filtered, with minimum performances of 0.74–0.84 of the best. At the lower end (minimum performances < 0.50 of the top classifier) were SIMCA, base and *ReliefF*-filtered *J48* classifiers, *HyperPipes*, *SimpleCart*, *NaiveBayes* and OSC-PLSDA (Fig. 2B). SIMCA would have been promoted to 27th rank if forced to make classifications (see Section 3.4.1).

4. Discussion

The data sets in this paper were purposely selected to be difficult classification challenges. It was expected that training and external test data would be non-IID, while the four projects presented different class structures, with either two, three, four, or five classes, and sometimes unbalanced numbers of samples in the classes. We evaluated classifiers in several projects in consideration of the unlikelihood of a learning algorithm with optimal performance for all tasks [20,55].

Currently, the most common validation in chemometrics is CV (Table 1), in which training and test samples are withdrawn from a unified data pool containing all the samples in the study. We found the consequences of departure from this standard protocol were striking. Validations by external test populations (EV) were only 50–80% as accurate as CV. Because we intentionally used external test data with potential sources of variation (Table 3), this underperformance in EV might be pessimistic, but in practice comparable studies are scarce (Table 1). Nonetheless, the EV challenges were feasible, as the best (among 28) classifiers achieved mean accuracies of 86–97% (kappa 0.71–0.97).

It is pertinent to review the tested classifiers in the context of their current status in analytical chemistry. Counts of papers, from the past decade, using relevant generic methods are in Table 7, with references in Supplementary Data Table S2. The fact that random forests featured in only 4.5% of the surveyed papers is noteworthy, as in our tests no criteria revealed statistical superiority to *RandomForest* of any other classifier. *RandomForest* was a top-ranked classifier in CV, and in EVs was in the top group by all criteria, i.e., rankings (Section 3.4.1), permutation (Section 3.4.2), and minimum

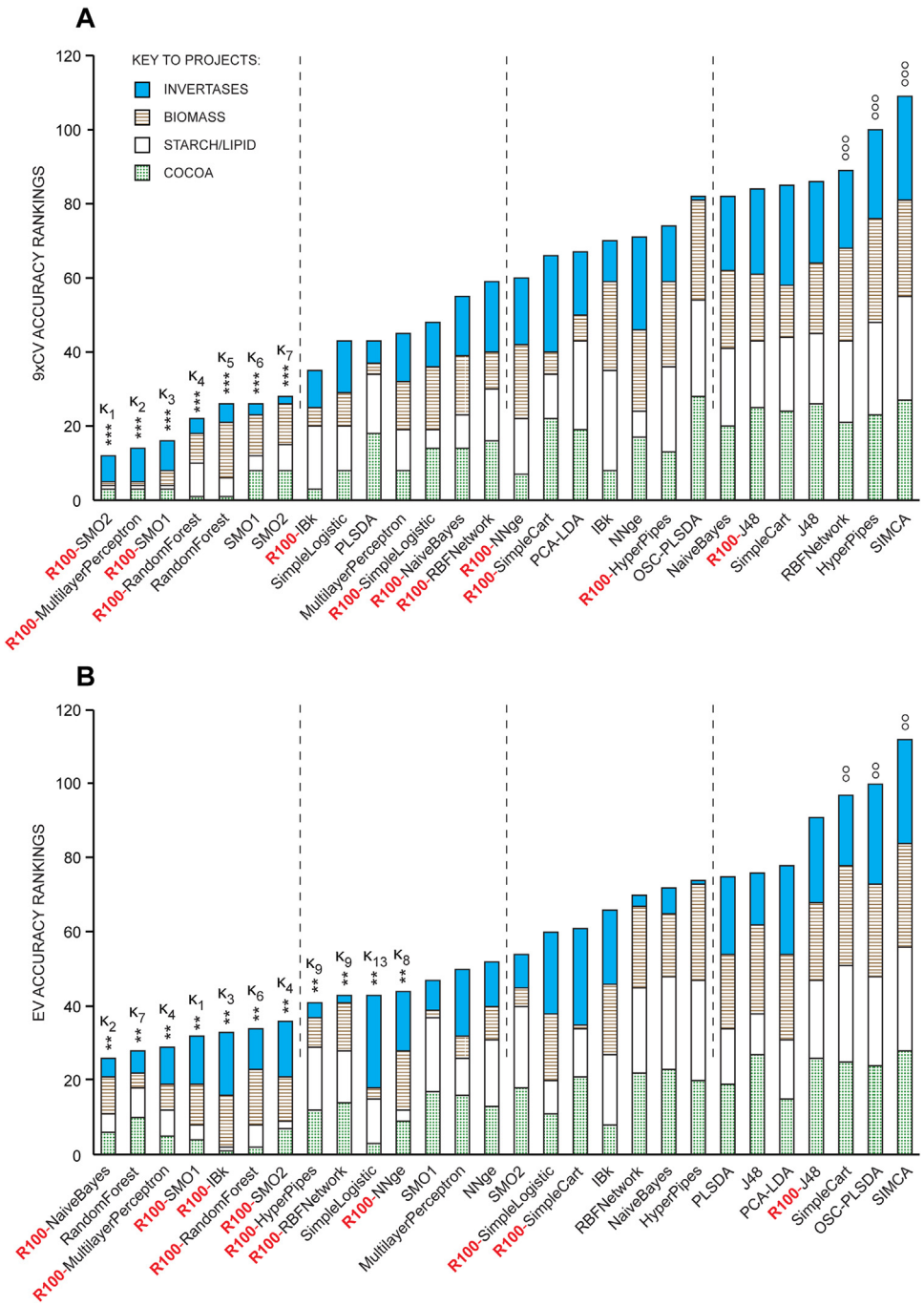


Fig. 1. Cumulative rankings of 28 classifiers by accuracy over the four projects, in (A) cross-validations (9 × CV), and (B) external validations (EV). 'R100' prefix: classifiers with prior selection of 100 features using *ReliefF*. 'SMO1', 'SMO2': *SMO* classifiers with linear or second-order polynomial kernels, respectively. Asterisks: 'top group' of classifiers not differing significantly (Friedman–Nemenyi tests, $^{**}P < 0.01$; $^{***}P < 0.001$) from the first rank. Open circles conversely define 'bottom groups'. Subscript of κ symbols: rankings by kappa among the top group. Vertical dotted lines: rank quartiles.

relative performance (Section 3.4.3). Its worst mean performance was 0.86 of the best classifier (on the Cocoa data), and this criterion placed *RandomForest* first among all classifiers.

Supplementary material related to this article can be found, in the online version, at http://dx.doi.org/10.1016/j.aca.2013.09.027.

RandomForest, moreover, outperformed the current favorite in chemometrics, PLSDA, which featured in nearly half of all reviewed papers (Table 7). One or more latent variable methods were used in 64% of papers, so it is surprising that all four such methods were in the bottom quartile of EV performances by the 28 classifiers. The best latent variable methods achieved only about two-thirds of the significant classifications by the top machine learning classifiers. OSC, which featured in less than 5% of papers, proved counterproductive in combination with PLSDA.

In theory, alternative data pre-processing or model construction may provide scope for improving classifiers for a given task [45,56]. Other common protocols for data scaling or optimization of model components were not, however, found to promote latent variable methods beyond third quartile in EV. Though improved solutions may always exist, a pragmatic consideration would be that other classifiers performed better without needing further investigation. Moreover, latent variable classifiers were relatively worse in EV than CV, which implies that optimization in training may not translate into improved performance on non-IID test data.

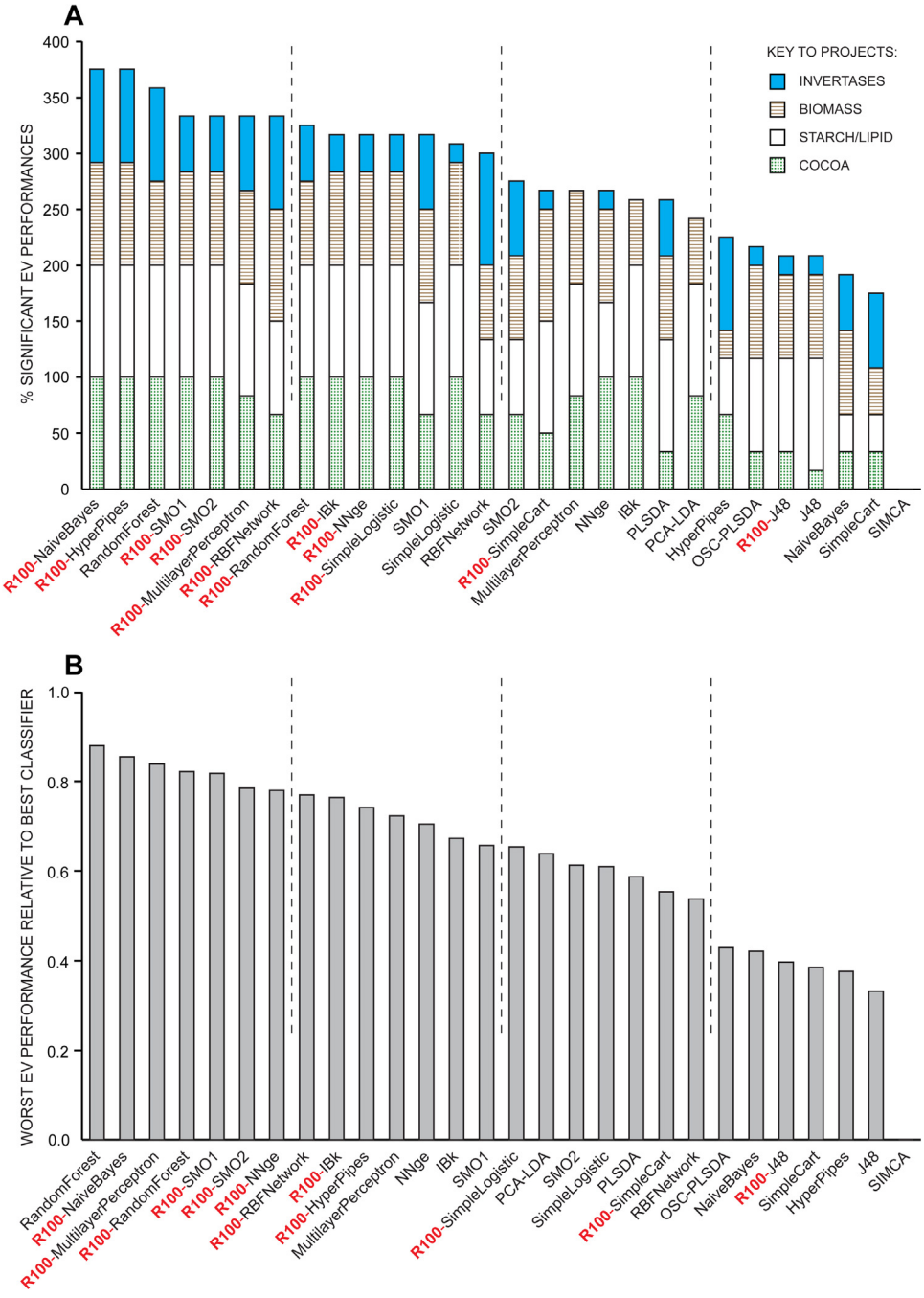


Fig. 2. Quality metrics for 28 classifiers over the four projects. (A) Percentage of predictions in each project ($n = 6–30$) found significant ($P < 0.05$) in permutation tests. (B) Each classifier's *worst* mean performance ($n = 6–30$) in any project, as a proportion of the best classifier performance on that data. Classifier labels as Fig. 1. Vertical dotted lines: rank quartiles.

Latent variable methods can be categorized as 'generative', since their models involve representations that could, in theory, be used to reconstruct realistic data. It has been suggested that, in practice, differences between generative models and the true data distribution mean their generalization is often poorer than purely 'discriminative' methods [57,58]. On this interpretation, the generative methods in this paper were vulnerable to the divergences between training and test data distributions, which were more pronounced in EV, and only exacerbated by OSC. This would be particularly applicable to the failure in EV of class-modeling by SIMCA, despite use of this method in 14% of surveyed papers (Table 7). A representative reference set is critical for SIMCA [59], which is vulnerable to data outliers [60].

In consequence, latent variable methods did not fully deliver their perceived benefits. The attractions of latent variables, in reduction of dimensionality while encapsulating coordinated behavior of multiple variables, remain for fields such as systems biology [61], while class-modeling approaches such as SIMCA can be particularly appropriate for situations such as quality control [21]. Nonetheless, it appears machine learning methods ultimately have the greater power for challenging classifications.

Among machine learning methods, one or both of neural networks and support vector machines were used in about one-third of papers surveyed (Table 7). An issue with these methods is that various model parameters can be adjusted, so a study with no tuning strategy might not do them full justice. We used

Table 7
Counts of papers using classifier approaches tested in this study, over ten years (2002–2012).[a]

Methodology	Number of papers	Representative(s) in this study
PLSDA	132	PLSDA
Neural networks	61	*MultilayerPerceptron*
Support vector machines	46	*SMO*
SIMCA	41	SIMCA
PCA-LDA	30	PCA-LDA
Nearest neighbors	29	*IBk*, *NNge*
Decision trees	17	*J48*, *SimpleCart*
Bayesian	14	*NaiveBayes*
OSC	13	OSC-PLSDA
Random forests	13	*RandomForest*
Logistic regression	6	*SimpleLogistic*
Radial basis function networks	4	*RBFNetwork*
Boosting	3	*SimpleLogistic*
HyperPipes	0	*HyperPipes*

[a] Journals: *Anal. Chem.*, *Anal. Chim. Acta*, *Chemom. Intell. Lab. Syst.*, *Food Chem.*, *Metabolomics*. Total number of papers: 286. See Table S2 in Supplementary Data for references.

the Weka default settings for parameters of *SMO* models (Supplementary data, Table S1). The same applied to *MultilayerPerceptron* models, except the number of hidden layers was set as the number of classes (Table S1). Neural network designs were therefore relatively simple, but took into account the nature of the classification task. Classifiers of both types were 100% accurate on virtually all training sets with the fixed settings. In $9 \times$ CV, moreover, the good performance of both methods justified their reputations.

Feature selection currently appears to be employed by a minority of machine learning studies in chemometrics. Among the 140 machine learning papers surveyed (Supplementary Data, Table S2), some form of feature selection was used by just over a quarter (26.4%), including 6.4% that demonstrated improved classification, and another 6.4% whose stated motivation was instead marker discovery or more interpretable models.

Feature selection significantly improved many of our machine learning classifiers. The *MultilayerPerceptron* and *SMO* classifiers were generally outperformed by their own extremely successful *ReliefF*-filtered versions. It is known that multilayer perceptrons are susceptible to high dimensionality [28,30], while support vector machines are not immune [62]. Improved performances due to feature selection have been noted in the chemometrics field for both these methodologies [63–65]. Although feature selection itself can be overfitted to training data [30], this did not emerge as a problem in the present study.

ReliefF-filtering also improved certain other EV classifiers (Friedman–Nemenyi tests; $P < 0.01$) for which high-dimensional data is considered challenging: *RBFNetwork* [28], *IBk* [30] and *NaiveBayes* [26]. Previous chemometrics reports on these generic classifiers have found feature selection to be beneficial [63,64,66]. The simple *HyperPipes*, though little used (Table 7), was also among the top EV classifiers when *ReliefF*-filtered.

High-dimensionality is also a problem for classical decision-tree algorithms [30], such as *J48* and *SimpleCart*, and improvement of decision trees by feature selection has been reported in chemometrics [63,64,66]. We did not find classical decision trees to be among the best performers even with filtering, though *SimpleCart* was significantly improved using *ReliefF* (Friedman–Nemenyi tests; $P < 0.01$). Poor model stability is another issue with the *J48* and *SimpleCart* algorithms; decisions about features near the root affect choices further down the tree, so small data variations can produce very different trees [25,67].

Dimensionality reduction is integral to the *SimpleLogistic* algorithm [20,29], and its performance was not improved by *ReliefF*. The *SimpleLogistic* base classifier, in fact, ranked in the top group by EV accuracy. This algorithm employs boosting and logistic regression [20], neither of which are widely used in chemometrics (Table 7).

The resilience of random forests to high dimensionality [27] was particularly evident in this study, since *ReliefF*-filtering did not improve *RandomForest* classifiers. *RandomForest* was the only base classifier (i.e., used without a feature selection filter) in the top group by all criteria in Sections 3.4.1, 3.4.2 and 3.4.3.

Although this study focused on class-assignment, additional utilities of random forests have recently been exploited in chemometrics. The fact that they use the totality of high-dimensional data can be advantageous if maximum data-driven knowledge is sought, since data pruning prior to classification may cause information loss [30]. In fact, fewer than half the 13 surveyed papers involving random forests (Supplementary Data, Table S2) used them purely for classification. Six papers instead used random forests for feature evaluation, via recursive feature elimination [68,69], or the algorithm's permutation-based 'importance scores' for every feature's contribution to classification [70–73]. Another obtained between-class 'margins', from the proportions of decision trees in the 'forest' voting for the correct and incorrect classes [74]. This last was among our own recent studies to exploit random forest margins [3,10,45,75].

While *RandomForest* appeared the 'winner' in our evaluations, we recommend it as a 'must-try' for chemometrics, not a sole 'classifier of choice'. Our results are primarily mean performances over several data sets within each project. They conceal the fact that in 10% of individual classification tasks within the projects, *RandomForest* was in the bottom quartile of the 28 classifiers. While not a bad statistic in context, this confirmed the existence of data structures for which random forests are not the best classifiers.

One should therefore be mindful of the discourse on whether a universal classifier for any data is attainable [55,76]. We found certain classifiers outstanding for one project, only to prove weak otherwise. *HyperPipes* was number 1 among 28 classifiers in EV of the two-class Invertases data, but ranked 20–28 for the other projects. Likewise, *ReliefF*-filtered *SimpleCart* had a remarkable mean EV accuracy of 96% on the 12 data sets of the three-class Biomass project, but only 38% and 46% on the Cocoa and Invertases data.

Future generations of classifiers for non-IID data may emerge from the 'transfer learning' concept [6,77]. This early-stage machine learning theme is predicated on the infeasibility for many real-world enterprises of maintaining classifier-training examples that perfectly reflect the population they need to query. Algorithms that explicitly generalize across data populations with different feature distributions would have great potential.

5. Conclusions

This study demonstrated that classification of analytical data has scope to progress in two major areas. First, in more challenging but realistic validation scenarios, which encompass the divergences between training and test samples that are inevitable if classifier schemes are to have real-world utility. The generally good performances in our CV routines dramatically deteriorated for the average classifier in EV. Second, evaluation studies need to identify the best classifiers for generalization to realistic test populations. Here, we found grounds for optimism in the superior EV performances among newer machine learning schemes. Those that fared best in EV were immune to, or actively reduced, the high dimensionality of spectrometric data. Random forests were the most successful representative of the former category, surpassing even the more popular support vector machines. Alternatively, feature selection by *ReliefF* proved successful, in combination with diverse

classifiers, both complex (e.g., neural networks) and simple (e.g., naive Bayes).

Acknowledgements

The authors acknowledge technical contributions from C. Atkinson, S. Cang, D. Enot, J. Heald, H. Johnson, D. Kell, B. Kular, C. Lu, R. Machado, C. Salt, S. Walsh and T. Wang. We thank Xuemin Wu for seed of the invertase mutants. The study was part funded by UK Biotechnology and Biological Sciences Research Council grants EGA17716 and BB/D006651/1.

References

[1] L.A. Berrueta, R.M. Alonso-Salces, K. Héberger, J. Chromatogr. A 1158 (2007) 196–214.
[2] K. Saito, F. Matsuda, Annu. Rev. Plant Biol. 61 (2010) 463–489.
[3] A.J. Lloyd, M. Beckmann, S. Haldar, C. Seal, K. Brandt, J. Draper, Am. J. Clin. Nutr. 97 (2013) 377–389.
[4] R. Madsen, T. Lundstedt, J. Trygg, Anal. Chim. Acta 659 (2010) 23–33.
[5] K.H. Esbensen, P. Geladi, J. Chemometr. 24 (2010) 168–187.
[6] B. Quanz, J. Huan, Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, 2009, pp. 1327–1336.
[7] J.E. Wood, D. Allaway, E. Boult, I.M. Scott, Anal. Chem. 82 (2010) 6048–6055.
[8] D.M. Hawkins, J. Kraker, J. Chemometr. 24 (2010) 188–193.
[9] L.A. Stone, R.W. Kennard, Technometrics 11 (1969) 137–148.
[10] I.M. Scott, C.P. Vermeer, M. Liakata, D.I. Corol, J.L. Ward, W. Lin, H.E. Johnson, L. Whitehead, B. Kular, J.M. Baker, S. Walsh, T.R. Larson, I.A. Graham, T.L. Wang, R.D. King, J. Draper, M.H. Beale, Plant Physiol. 153 (2010) 1506–1520.
[11] A. Rudnitskaya, I. Delgadillo, S.M. Rocha, A.-M. Costa, A. Legin, Anal. Chim. Acta 563 (2006) 315–318.
[12] M.S. Cosio, D. Ballabio, S. Benedetti, C. Gigliotti, Anal. Chim. Acta 567 (2006) 202–210.
[13] J. Luypaert, S. Heuerding, D.L. Massart, Y.V. Heyden, Anal. Chim. Acta 582 (2007) 181–189.
[14] J. Broséus, M. Vallat, P. Esseiva, Chemom. Intell. Lab. Syst. 107 (2011) 343–350.
[15] C.J. Bevin, R.G. Dambergs, A.J. Fergusson, D. Cozzolino, Anal. Chim. Acta 621 (2008) 19–23.
[16] P. Jonsson, J. Gullberg, A. Nordström, M. Kusano, M. Kowalczyk, M. Sjöström, T. Moritz, Anal. Chem. 76 (2004) 1738–1745.
[17] E.D. Dawson, C.L. Moore, D.M. Dankbar, M. Mehlmann, M.B. Townsend, J.A. Smagala, C.B. Smith, N.J. Cox, R.D. Kuchta, K.L. Rowlen, Anal. Chem. 79 (2006) 378–384.
[18] M.R. Viant, C. Ludwig, S. Rhodes, U.L. Günther, D. Allaway, Metabolomics 3 (2007) 453–463.
[19] T. Rajalahti, O.M. Kvalheim, Int. J. Pharm. 417 (2011) 280–290.
[20] I.H. Witten, E. Frank, Data Mining Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco, 2005.
[21] M. Forina, P. Oliveri, S. Lanteri, M. Casale, Chemom. Intell. Lab. Syst. 93 (2008) 132–148.
[22] A.M. Martínez, A.C. Kak, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2001) 228–233.
[23] M. Barker, W. Rayens, J. Chemometr. 17 (2003) 166–173.
[24] S. Wold, H. Antti, F. Lindgren, J. Ohman, Chemom. Intell. Lab. Syst. 44 (1998) 175–185.
[25] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, M. Steinbach, D. Hand, D. Steinberg, Knowl. Inf. Syst. 14 (2008) 1–37.
[26] T.A. Almeida, J. Almeida, A. Yamakami, J. Internet Serv. Appl. 1 (2011) 183–200.
[27] A. Liaw, M. Wiener, R News 2 (2002) 18–22.
[28] F. Marini, Anal. Chim. Acta 635 (2009) 121–131.
[29] N. Landwehr, M. Hall, E. Frank, Mach. Learn. 59 (2005) 161–205.
[30] C.F. Aliferis, A. Statnikov, I. Tsamardinos, Cancer Inform. 2 (2006) 133–162.
[31] M.A. Hall, G. Holmes, IEEE Trans. Knowl. Data Eng. 15 (2003) 1437–1447.
[32] E. Szymańska, E. Saccenti, A.K. Smilde, J.A. Westerhuis, Metabolomics 8 (2012) S3–S16.
[33] A. Ben-David, Expert Syst. Appl. 34 (2008) 825–832.
[34] D.M.W. Powers, Proceedings of the IEEE International Conference on Information Science and Technology, Hubei, China, 2012, pp. 567–573.
[35] J. Li, J.P. Fine, Biostatistics 9 (2008) 566–576.
[36] J. Demšar, J. Mach. Learn. Res. 7 (2006) 1–30.
[37] K. Héberger, Trends Anal. Chem. 29 (2010) 101–109.
[38] J. Lyons-Weiler, R. Pelikan, H.J. Zeh, D.C. Whitcomb, D.E. Malehorn, W.L. Bigbee, M. Hauskrecht, Cancer Inform. 1 (2005) 53–77.
[39] L. Xiang, K. Le Roy, M.R. Bolouri-Moghaddam, M. Vanhaecke, W. Lammens, F. Rolland, W. Van den Ende, J. Exp. Bot. 62 (2011) 3849–3862.
[40] S.M. Sherson, H.L. Alford, S.M. Forbes, G. Wallace, S.M. Smith, J. Exp. Bot. 54 (2003) 525–531.
[41] Z. Tymowska-Lalanne, M. Kreis, Planta 207 (1998) 259–265.
[42] I.M. Scott, S.M. Clarke, J.E. Wood, L.A.J. Mur, Plant Physiol. 135 (2004) 1040–1049.
[43] J.L. Ward, S. Forcat, M. Beckmann, M. Bennett, S.J. Miller, J.M. Baker, N.D. Hawkins, C.P. Vermeer, C. Lu, W. Lin, W.M. Truman, M.H. Beale, J. Draper, J.W. Mansfield, M. Grant, Plant J. 63 (2010) 443–457.
[44] J.C. Motamayor, P. Lachenaud, J.W. da Silva e Mota, R. Loor, D.N. Kuhn, J.S. Brown, R.J. Schnell, PLoS ONE 3 (2008) e3311.
[45] D.P. Enot, W. Lin, M. Beckmann, D. Parker, D.P. Overy, J. Draper, Nat. Protoc. 3 (2008) 446–470.
[46] W.N. Venables, B.D. Ripley, Modern Applied Statistics with S, Springer, New York, 2002.
[47] H.T. Eastment, W.J. Krzanowski, Technometrics 24 (1982) 73–77.
[48] E. Frank, M. Hall, L. Trigg, G. Holmes, I.H. Witten, Bioinformatics 20 (2004) 2479–2481.
[49] P. Smialowski, D. Frishman, S. Kramer, Bioinformatics 26 (2010) 440–443.
[50] R.G. Brereton, Trends Anal. Chem. 25 (2006) 1103–1111.
[51] K. Hornik, C. Buchta, A. Zeileis, Comput. Stat. 24 (2009) 225–232.
[52] A.-L. Boulesteix, K. Strimmer, Theor. Biol. Med. Model. 2 (2005) 23.
[53] Ø. Hammer, D.A.T. Harper, P.D. Ryan, Palaeontol. Electron. 4 (1) (2001).
[54] R.G. Brereton, J. Chemometr. 25 (2011) 225–246.
[55] D.H. Wolpert, The supervised learning no-free-lunch theorems, in: R. Roy, M. Koppen, S. Ovaska, T. Furuhashi, F. Hoffman (Eds.), Soft Computing and Industry: Recent Applications, Springer-Verlag London Ltd., Godalming, 2002, pp. 25–42.
[56] R.G. Brereton, Chemometrics for Pattern Recognition, Wiley, Chichester, 2009.
[57] C.M. Bishop, J. Lasserre, in: J.M. Bernardo, M.J. Bayarri, J.O. Berger, A.P. Dawid, D. Heckerman, A.F.M. Smith, M. West (Eds.), Bayesian Statistics 8, Oxford University Press, Oxford, 2007, pp. 3–24.
[58] G. Bouchard, B. Trigg, in: J. Antoch (Ed.), COMPSTAT 2004 – Proceedings in Computational Statistics, Physica Verlag, Heidelberg, 2004, pp. 697–704.
[59] G.R. Flåten, B. Grung, O.M. Kvalheim, Chemom. Intell. Lab. Syst. 72 (2004) 101–109.
[60] K. Vanden Branden, M. Hubert, Chemom. Intell. Lab. Syst. 79 (2005) 10–21.
[61] M. Ringnér, Nat. Biotechnol. 26 (2008) 303–304.
[62] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Mach. Learn. 46 (2002) 389–422.
[63] J. Boccard, A. Kalousis, M. Hilario, P. Lantéri, M. Hanafi, G. Mazerolles, J.-L. Wolfender, P.-A. Carrupt, S. Rudaz, Chemom. Intell. Lab. Syst. 104 (2010) 20–27.
[64] D. Prilutsky, E. Shneider, A. Shefer, B. Rogachev, L. Lobel, M. Last, R.S. Marks, Anal. Chem. 83 (2011) 4258–4265.
[65] J. Schmitt, M. Beekes, A. Brauer, T. Udelhoven, P. Lasch, D. Naumann, Anal. Chem. 74 (2002) 3865–3868.
[66] G. Van Dijck, M.M. Van Hulle, Chemom. Intell. Lab. Syst. 107 (2011) 318–332.
[67] B. Briand, G.R. Ducharme, V. Parache, C. Mercat-Rommens, Comput. Stat. Data Anal. 53 (2009) 1208–1217.
[68] D. Donald, T. Hancock, D. Coomans, Y. Everingham, Chemom. Intell. Lab. Syst. 82 (2006) 2–7.
[69] P.M. Granitto, C. Furlanello, F. Biasioli, F. Gasperi, Chemom. Intell. Lab. Syst. 83 (2006) 83–90.
[70] L. Auret, C. Aldrich, Chemom. Intell. Lab. Syst. 105 (2011) 157–170.
[71] X. Lin, Q. Wang, P. Yin, L. Tang, Y. Tan, H. Li, K. Yan, G. Xu, Metabolomics 7 (2011) 549–558.
[72] X. Tang, J. Xiao, Y. Li, Z. Wen, Z. Fang, M. Li, Chemom. Intell. Lab. Syst. 118 (2012) 317–323.
[73] T. Tynkkynen, J. Mursu, T. Nurmi, K. Tuppurainen, R. Laatikainen, P. Soininen, Metabolomics 8 (2010) 386–398.
[74] D.P. Enot, J. Draper, Metabolomics 3 (2007) 349–355.
[75] M. Beckmann, D.P. Enot, D.P. Overy, I.M. Scott, P.G. Jones, D. Allaway, J. Draper, Br. J. Nutr. 103 (2010) 1127–1138.
[76] E. Barnard, Neural Comput. 23 (2011) 1–11.
[77] S.J. Pan, Q. Yang, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359.